Quiz 4

Started: Nov 10 at 12:13pm

Quiz Instructions

Open book;

You can use as much time as you want before the deadline, but once you submitted you cannot re-submit;

You cannot communicate with other people during the exam;

You are allowed to use webpage browsers or other apps during the exam;

Feel free to contact me or TA for any questions during the exam.

Questions
- Question 1
- **Question 2**
- Question 3
- **Question 4**
- **Question 5**
- Question 6
- **Question 7**
- Question 8

Time Elapsed: Hide Time
Attempt due: Nov 10 at 11:59pm
9 Hours, 0 Minutes, 49 Seconds

Question 1 — 1 pts

Which statement is false regarding Prim and Kruskal's algorithm? [Note: We use m to denote the number of edges, n to denote the number of vertices]

- (•) In Prim's algorithm, the graph must be connected for the algorithm to work, while Kruskal's algorithm can generate a minimum spanning forest in a disconnected graph.
- ( ) Prim's algorithm typically requires a priority queue data structure to select the next minimum edge, whereas Kruskal's algorithm requires sorting of all edges.
- ( ) The worst-case time complexity for both Prim's and Kruskal's algorithms is O(mn) when simple data structures are used.
- ( ) Both Prim's and Kruskal's algorithms can handle negative edge weights as long as the graph does not contain any cycles.

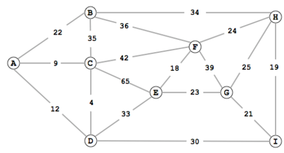
Question 2 — 1 pts

Let *T* be any spanning tree of a connected graph *G* with *n* vertices and *m* Which of the following statements **MUST** be true?

- (•) T has at least one leaf.
- ( ) Every cutset of G that disconnects G into two subgraphs must contain at least one edge from T. (A cutset is a set of edges whose removal from a graph increases the number of connected components.)
- ( ) If G is not a complete graph, then T will have fewer edges than G.
- ( ) T can be formed by removing exactly n−1 edges from G.

Question 3 — 1 pts

Which is the wrong statement about this graph?



- (•) An edge with the highest weight cannot be part of any minimum spanning tree.
- ( ) Edge (E,F) must be in the minimum spanning tree.
- ( ) If removing an edge increases the number of connected components in the graph, that edge must be part of every minimum spanning tree.
- ( ) Edge (C,D) must be in the minimum spanning tree.

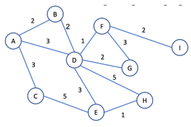
Question 4 — 1 pts

Which is wrong about shortest path search?

- ( ) Bellman-Ford algorithm can be used to detect negative weight cycles in a graph.
- ( ) A* search algorithm requires an admissible heuristic to guarantee the shortest path.
- ( ) Breadth-First Search (BFS) is guaranteed to find the shortest path in a weighted graph if all edge weights are equal.
- (•) Dijkstra's algorithm can handle graphs with negative edge weights if all cycles have a non-negative total weight.

Question 5 — 1 pts

Given the following weighted graph, which of the following statements is correct?



- ( ) Using Prim's algorithm starting from vertex A, edge (A,D) will necessarily be included in the MST.
- ( ) The sum of the weights in any MST of this graph will not be greater than the sum of the weights of the shortest path from A to I.
- (•) The edge with the least weight, which is (E,H), will necessarily be included in every MST of this graph.
- ( ) If Kruskal's algorithm is used to find the minimum spanning tree (MST), edge (F,G) will be included before edge (C,E).

Question 6 — 1 pts

Examplify at least two applications of minimum spanning trees (MST). And explain why MST is preferred in these applications.

Edit View Insert Format Tools Table
12pt Paragraph B I U A T²

Minimum spanning trees can be useful in network design like telecommunication network. It enables the most efficient delivery of information in a connected structure. It could also be used in circuit design, for example in electronic circuit design. It can maximum efficiency while connecting all the component together.

p 48 words

Question 7 — 1 pts

In the context of the union-find data structure, consider a sequence of *m* **union** and **find** operations on *n* Which of the following statements is true?

- ( ) Applying path compression to only union operations instead of find can lead to a linear time complexity for find over the sequence of operations.
- (•) Weighted-union without path compression guarantees that the amortized time complexity for each operation is O(log n).
- ( ) Path compression alone, without weighted-union, can result in a worse-case time complexity of O(n) for a single find operation.
- ( ) With both weighted-union and path compression applied, the amortized time complexity per operation is O(α(n)), where α is the inverse Ackermann function, which grows very slowly.

Question 8 — 1 pts

Given a set of vertices labeled from 0 to 9, a series of union operations are performed. After these operations, what is the number of connected components that include at least three vertices?

2-4 3-4 5-6 5-8 0-9 1-9 1-7 3-8 2-5

- ( ) 1
- ( ) 4
- ( ) 3
- (•) 2

Quiz saved at 9:14pm Submit Quiz